Análise de séries temporais

por similaridade e alinhamento não linear com Dynamic Time Warping

PETROBRAS ufscar DC MIDAS

Busca por Similaridade com DTW

Provendo eficiência

Eficiência computacional

O termo “eficiência” é bastante genérico e pode se referir a tempo de execução, memória, etc

Na Computação, usamos a análise assintótica como ferramenta

- Mas vamos pular detalhes

DE vs DTW

Vamos considerar duas séries temporais de mesmo comprimento *n*.

A distância euclidiana faz uma operação para cada par de observação

$$ed(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

~ 3 · n operações

Portanto, o número de operações realizadas é proporcional ao tamanho da entrada (do algoritmo), que no caso é *n*.

Dizemos, portanto, que a ED é O(n)[1] ou que o algoritmo é linear.

↳ big-oh

1. Tem muita simplificação aqui, mas aceitamos assim, mesmo na Computação

DE vs DTW

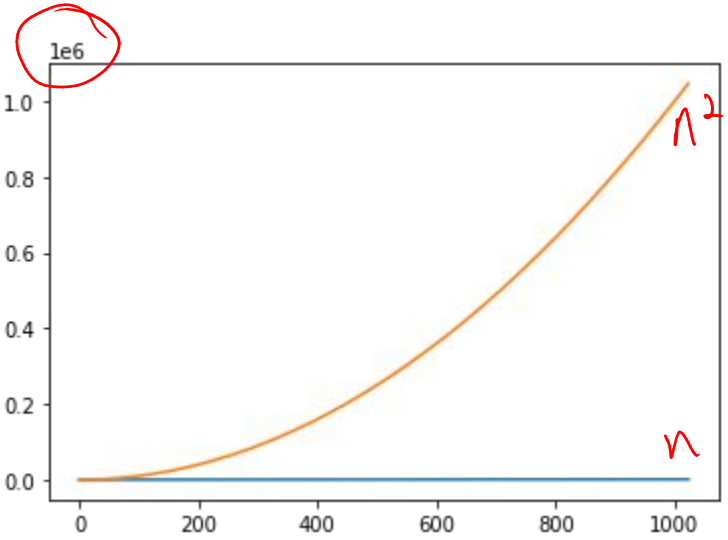
A DTW precisa preencher a matriz de custo, que é $n \times n$. Ou seja, a DTW precisa fazer $O(n^2)$ operações.

- Portanto, dizemos que a DTW é quadrática (nesse caso, tanto em tempo de execução quanto em memória)
- Usando as bandas de Sakoe-Chiba de tamanho w, ficamos com $O(n \cdot w)$

Isso é muito importante, pois o tempo de execução depende do número de operações.

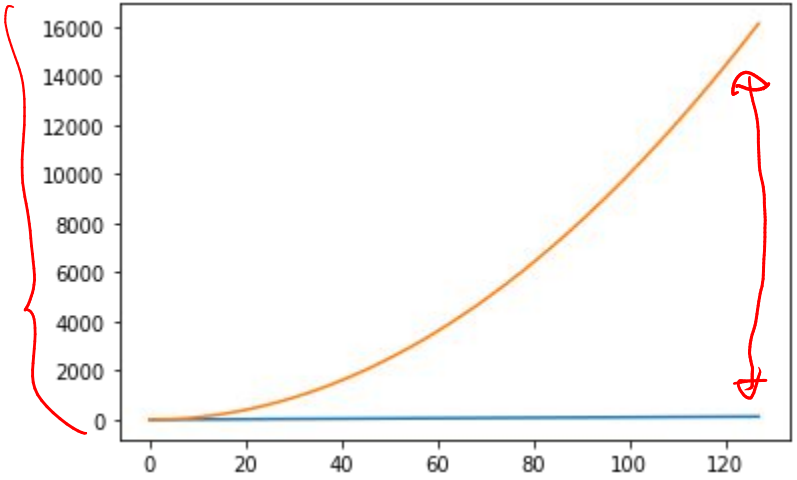
Linear vs Quadrático

n = 1024



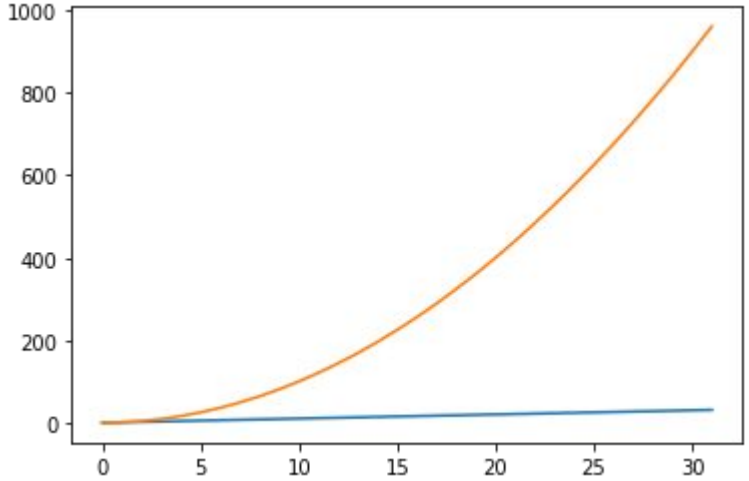
Linear vs Quadrático

n = 128



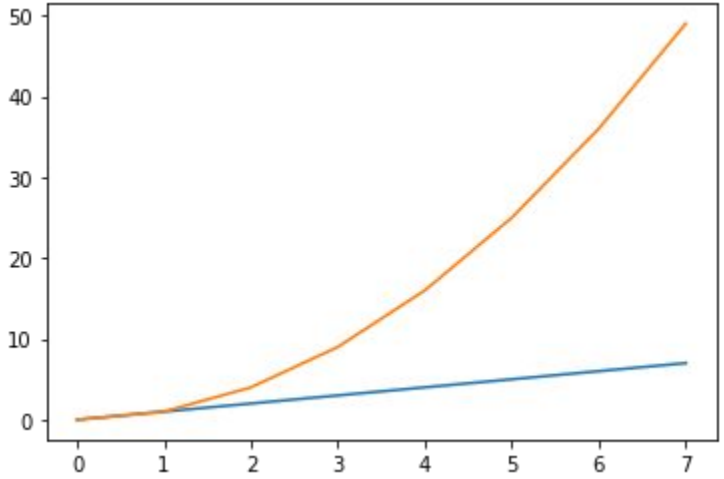
Linear vs Quadrático

n = 32



Linear vs Quadrático

n = 8



DE vs DTW

Para comparar um único par de séries temporais, a diferença não é significativa, na prática.

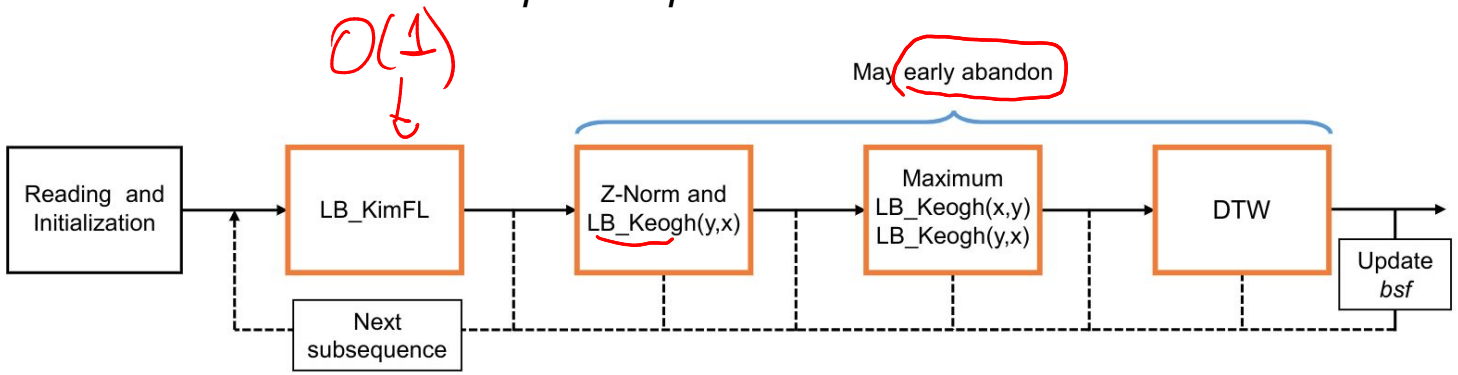
Mas e se eu for comparar vários, como na busca por similaridade?

- O valor *n* é o tamanho da consulta
- Multiplica-se um fator *l*, que é o comprimento da série de referência
  - e *l* costuma ser bem grande (esse é exatamente o objetivo)

$O(l \cdot complex(dist))$

UCR Suite

Portanto, para **escalar** uma busca por similaridade baseada em DTW, é preciso de várias técnicas de *speed-up*.



Um ponto importante antes de dissecar o pipeline: *best-so-far*

Silva, D. F., Giusti, R., Keogh, E., & Batista, G. E. (2018). Speeding up similarity search under dynamic time warping by pruning unpromising alignments. Data Mining and Knowledge Discovery, 32(4), 988-1016.
UCR USP Suite

UCR Suite

O primeiro ponto importante é não ~~tirar~~ calcular a raiz da DTW

- Sim, eu dei uma mentida até aqui
- ED e DTW tem uma raiz quadrada e eu omiti a do DTW

Motivo...

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

iteração 500

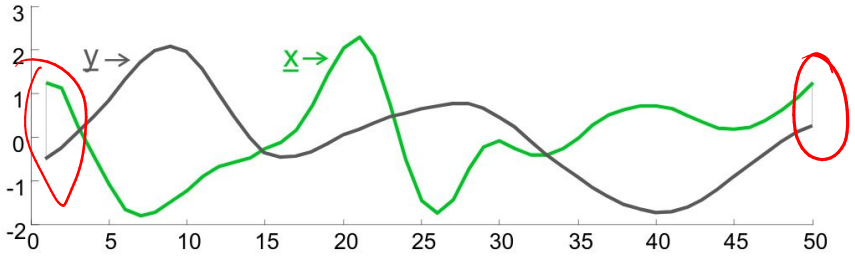
d(bsf) = 7,45

bsf = best-so-far

O(1)

LB_Kim_FL(500) = 8,49

LB-Kim-FL



Um *lower bound* é uma medida garantidamente menor que a *DTW distance*.

- Para ser útil, precisa ser rápido de ser calculado

$$LB\_Kim\_FL = (x_1-y_1)^2+(x_n-y_n)^2 \leq DTW(X,Y)$$

Considerando-se a indexação começando em 1

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

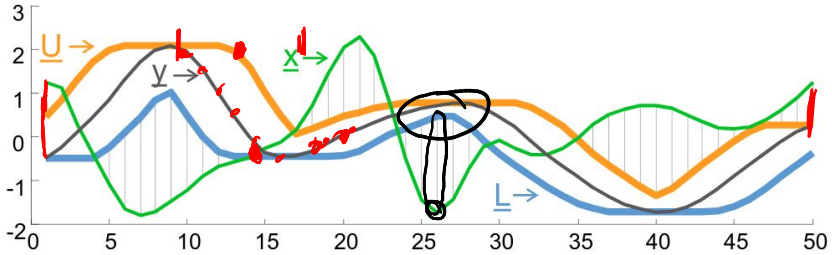
LB-Keogh

A *suite* considera LB_Keogh(X,Y) e LB_Keogh(Y,X)

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

LB-Keogh

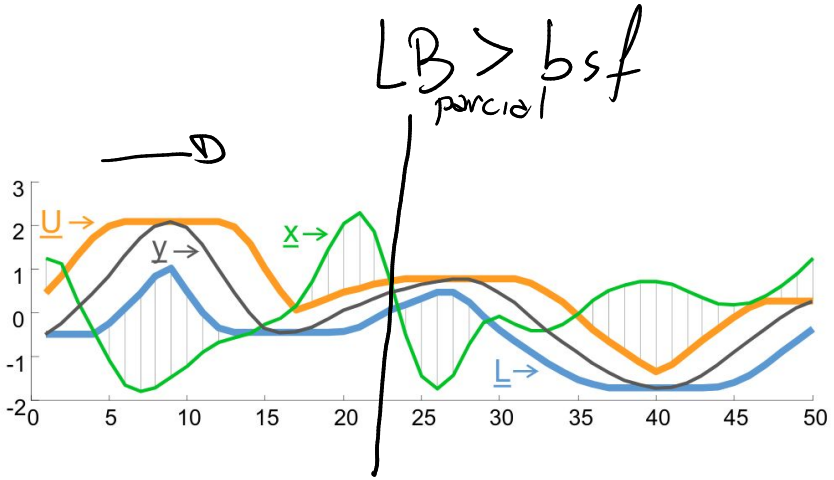


A *suite* considera LB_Keogh(X,Y) e LB_Keogh(Y,X)

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

LB-Keogh



A *suite* considera LB_Keogh(X,Y) e LB_Keogh(Y,X)

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

Early abandon do lower bound

- Ao calcular o LB-Keogh, é possível abandonar precocemente
- Se o cálculo parcial ultrapassa o *bsf*, podemos descartar a subsequência

Early abandon da z-normalização

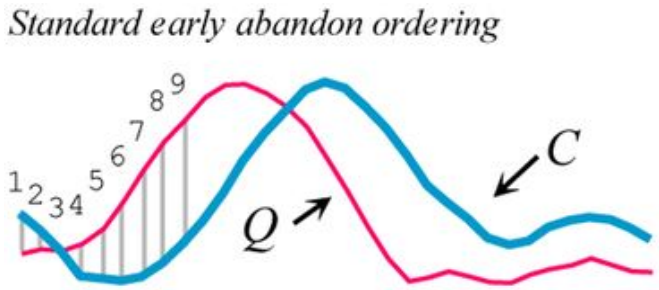
- Considera-se a busca sempre z-normalizada por subsequência
- Em vez de calcular de início, alterna-se com o LB

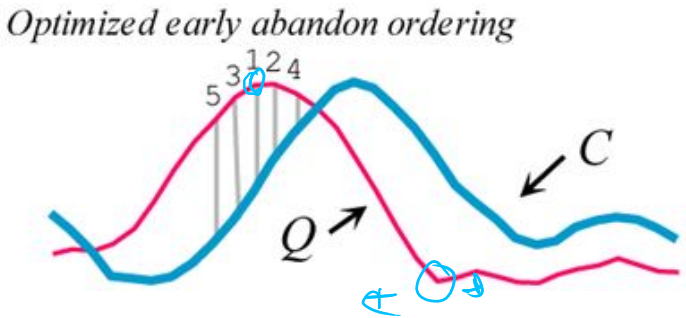
+ early abandon da DTW

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

UCR Suite

Re-ordenar o cálculo





Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012, August). Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 262-270).

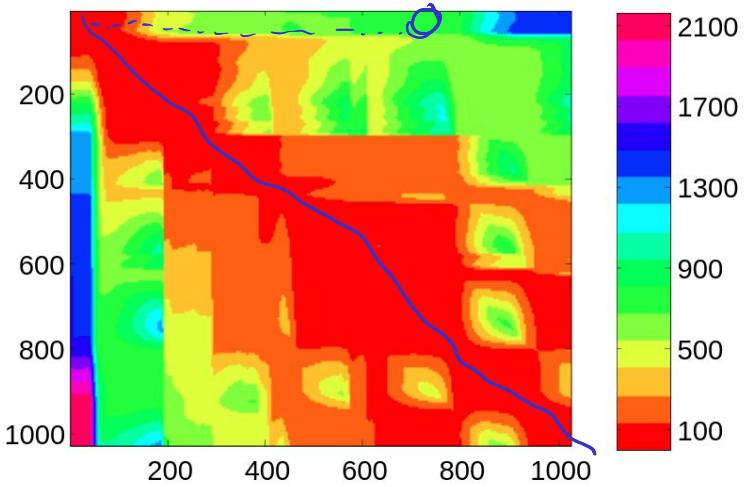
UCR-USP Suite

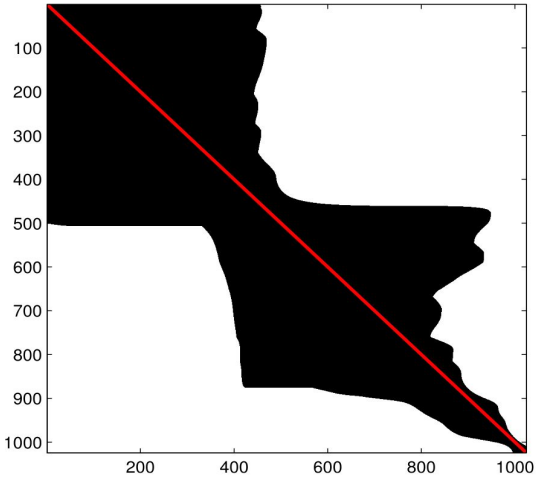
UB = dist. Euclidiana

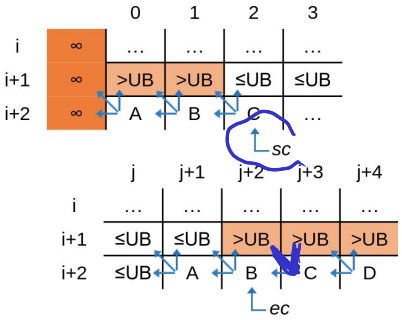
↳ na busca: UB = bsf

Pruned DTW

Mais otimizações







Silva, D. F., & Batista, G. E. (2016, June). Speeding up all-pairwise dynamic time warping matrix calculation. In Proceedings of the 2016 SIAM International Conference on Data Mining (pp. 837-845). Society for Industrial and Applied Mathematics.

UCR-USP Suite

Vamos implementar isso tudo? Não, claro. Mas vamos à prática